

Application
Number

IDS Flag Clearance for Application 10623099

**IDS
Information**

Content	Mailroom Date	Entry Number	IDS Review	Last Modified	Reviewer
M844	2005-10-03	16	Y <input checked="" type="checkbox"/>	2006-04-04 11:37:05.0	MLe
M844	2003-07-18	13	Y <input checked="" type="checkbox"/>	2006-04-04 11:37:04.0	MLe
M844	2005-10-03	10	Y <input checked="" type="checkbox"/>	2006-04-04 11:37:04.0	MLe

10/623,099


 Home | Login | Logout | Access Information | Alerts | Sitemap | Help
 Welcome United States Patent and Trademark Office

 Search Session History

BROWSE

SEARCH

IEEE Xplore Guide

SUPPORT

Thu, 21 Sep 2006, 10:16:48 PM EST

 Edit an existing query or
 compose a new query in the
 Search Query Display.

Search Query Display


Select a search number (#)
 to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Recent Search Queries

		Results
#1	((fault-tolerant<in>metadata) <and> (non-fault-tolerant<in>metadata))<and> (redundancy check<in>metadata)	0
#2	((fault-tolerant<in>metadata) <and> (non-fault-tolerant<in>metadata))<and> (redundancy<in>metadata)	4
#3	((fault-tolerant<in>metadata) <and> (non-fault-tolerant<in>metadata))<and> (redundancy<in>metadata)	4
#4	((fault-tolerant<in>metadata) <and> (non-fault-tolerant<in>metadata))<and> (redundancy<in>metadata)	4
#5	((fault-tolerant<in>metadata) <and> (non-fault-tolerant<in>metadata))<and> (redundancy<in>metadata)	4
#6	((fault-tolerant<in>metadata) <and> (non-fault-tolerant<in>metadata))<and> (redundancy<in>metadata)	4
#7	((redundant<in>metadata) <and> (non-redundant<in>metadata))<and> (fault-tolerant<in>metadata)	9
#8	((redundant<in>metadata) <and> (non-redundant<in>metadata))<and> (fault-tolerant<in>metadata)	9
#9	((redundant<in>metadata) <and> (non-redundant<in>metadata))<and> (fault-tolerant<in>metadata)	9
#10	((redundant<in>metadata) <and> (non-redundant<in>metadata))<and> (fault-tolerant<in>metadata)	9
#11	((redundant<in>metadata) <and> (non-redundant<in>metadata))<and> (fault-tolerant<in>metadata)	9
#12	((decode<in>metadata) <and> (non-redundant<in>metadata))<and> (fault-tolerant<in>metadata)	0


[Help](#) [Contact Us](#) [Privacy & Security](#) [IEEE.org](#)

© Copyright 2006 IEEE – All Rights Reserved

10/623,099


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
[Search: The ACM Digital Library](#) [The Guide](#)

fault-tolerant and non-fault-tolerant and redundancy checking

SEARCH
[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

[fault tolerant](#) and [non fault tolerant](#) and [redundancy checking](#) and [without redundancy check](#) and [single CPU](#) and [distinct inst](#)
Sort results by [relevance](#)
[Save results to a Binder](#)
[Try an Advanced Search](#)Display results [expanded form](#)
[Search Tips](#)
[Try this search in The ACM Guide](#)
[Open results in a new window](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance

1 [Understanding fault-tolerant distributed systems](#)

Flavin Cristian

February 1991 **Communications of the ACM**, Volume 34 Issue 2

Publisher: ACM Press

Full text available: [pdf\(6.17 MB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)**2** [Reliability Issues in Computing System Design](#)

B. Randell, P. Lee, P. C. Treleaven

June 1978 **ACM Computing Surveys (CSUR)**, Volume 10 Issue 2

Publisher: ACM Press

Full text available: [pdf\(3.95 MB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**3** [Emerging areas: Fault-tolerant platforms for automotive safety-critical applications](#)

M. Baleani, A. Ferrari, L. Mangeruca, A. Sangiovanni-Vincentelli, Maurizio Peri, Saverio Pezzini

October 2003 **Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems**

Publisher: ACM Press

Full text available: [pdf\(736.40 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Fault-tolerant electronic sub-systems are becoming a standard requirement in the automotive industrial sector as electronics becomes pervasive in present cars. We address the issue of fault tolerant chip architectures for automotive applications. We begin by reviewing fault-tolerant architectures commonly used in other industrial domains where fault-tolerant electronics has been a must for a number of years, e.g., the aircraft manufacturing industrial sector. We then proceed to investigate how to ...

Keywords: VLSI, automotive, fault-tolerant, multi-processor, safety critical, system-on-a-chip

4 [Transient fault detection via simultaneous multithreading](#)

Steven K. Reinhardt, Shubhendu S. Mukherjee

May 2000 **ACM SIGARCH Computer Architecture News, Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: The ACM Digital Library The Guide

fault-tolerant and non-fault-tolerant and redundancy checking


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

[fault tolerant](#) and [non fault tolerant](#) and [redundancy checking](#) and [without redundancy check](#) and [single CPU](#) and [distinct inst](#)

 Sort results by
 [Save results to a Binder](#)
[Try an Advanced Search](#)

 Display results
 [Search Tips](#)
[Try this search in The ACM Guide](#)
 [Open results in a new window](#)

Results 181 - 200 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) **10**

Best 200 shown

Relevance

181 [A novel approach to accurate timing verification using RTL descriptions](#)

K. Roy, J. A. Abraham

 June 1989 [Proceedings of the 26th ACM/IEEE conference on Design automation](#)
Publisher: ACM Press

 Full text available: [pdf\(515.50 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Timing verification is a critical part of VLSI circuit design. A new approach to timing verification using Register Tr Level (RTL) descriptions is presented, which eliminates false paths that occur due to (i) redundancy, (ii) reconver fanout or (iii) control signal constraints, and generates a test for the critical paths. High level instructions of the c used to test for any timing violations. An algorithm to identify a ...

182 [Special session on reconfigurable computing: Designing and testing fault-tolerant techniques for SRAM-based FPGAs](#)

Fernanda Lima Kastensmidt, Gustavo Neuberger, Luigi Carro, Ricardo Reis

 April 2004 [Proceedings of the 1st conference on Computing frontiers](#)
Publisher: ACM Press

 Full text available: [pdf\(390.51 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper discusses fault-tolerant techniques for SRAM-based FPGAs. These techniques can be based on circuit modifications, with obvious modifications in the programmable architecture, or they can be implemented at the high-level description, without modification in the FPGA architecture. The high-level method presented in this work is based on Modular Redundancy (TMR) and a combination of Duplication Modular Redundancy (DMR) with Concurrent Error (CED) techniques, which ...

Keywords: FPGA, fault-tolerance

183 [Live-structure dataflow analysis for Prolog](#)

Anne Mulkers, William Winsborough, Maurice Bruynooghe

 March 1994 [ACM Transactions on Programming Languages and Systems \(TOPLAS\)](#), Volume 16 Issue 2

Publisher: ACM Press

 Full text available: [pdf\(3.59 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

For the class of applicative programming languages, efficient methods for reclaiming the memory occupied by re structures constitute an important aspect of current implementations. The present article addresses the problem reuse for logic programs through program analysis rather than by run-time garbage collection. The aim is to derive time properties that can be used at compile time to specialize the target code for a program according to a given

Interficon S, 10/623,099

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	642	(714/11).ccls.	US-PGPUB; USPAT	OR	ON	2006/09/21 22:32
L2	269	(714/12).ccls.	US-PGPUB; USPAT	OR	ON	2006/09/21 22:33
L3	522	(714/13).ccls.	US-PGPUB; USPAT	OR	ON	2006/09/21 22:33
L4	16302	fault-toleran\$4 or (fault adj toleran\$4)	US-PGPUB; USPAT	OR	ON	2006/09/21 22:34
L6	19910	fault-toleran\$4 or (fault adj toleran\$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/21 22:35
L7	104	decode same (L6 or non-fault-toleran\$4 or (non adj fault adj toleran\$4))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/21 22:35
L8	11	L7 same (redundan\$4 or (without adj redundan\$4) or (non-redundan\$4) or (non adj redundan\$4))	US-PGPUB; USPAT	OR	ON	2006/09/21 22:35
L9	104	decode same (L6 or non-fault-toleran\$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/21 22:36
L10	11	L9 same (redundan\$4 or (without adj redundan\$4) or (non-redundan\$4))	US-PGPUB; USPAT	OR	ON	2006/09/21 22:36
L11	2	7 same (compar\$4 or match\$4)	US-PGPUB; USPAT	OR	ON	2006/09/21 22:38
L12	0	9 and (1 or 2 or 3) and "single cpu"	US-PGPUB; USPAT	OR	ON	2006/09/21 22:41